Received: 10 March 2023 | Revised: 14 August 2023 | Accepted: 16 August 2023

DOI: 10.1002/sys.21721

WILEY

REGULAR ARTICLE

Model-based standards authoring: ISO 15288 as a case in point

Dov Dori

Faculty of Data and Decision Sciences, Technion, Israel Institute of Technology, Haifa, Israel

Correspondence
Dov Dori, Faculty of Data and Decision Sciences, Technion, Israel Institute of Technology, Haifa, Israel.
Email: dori@technion.ac.il

Abstract

ISO/IEC/IEEE 15288:2015 is one of the most fundamental systems engineering international standards. In this work, the major system lifecycle processes specified in 15288 and, equally importantly, the objects interacting through them, are modeled meticulously using OPM ISO 19450. The conceptual model, based on this standard's text, reflects the implied authors' intent, bringing up ambiguities that arise from the informality of natural language text and reference to related figures. The resulting OPM model is an exact, formal, and detailed expression of the processes and related objects in the first part of 15288, making it machine interpretable. The gaps discovered during the modeling process are testimony to the value of the model-based standards authoring approach and the centrality of a formal yet humanly accessible model as the underlying backbone of international standards and key technical documents in general.

KEYWORDS
conceptual modeling, ISO/IEC/IEEE 15288, model-based standards authoring, model-based systems engineering, OPM ISO 19450

1 | INTRODUCTION

Functioning to shape up mutual understanding and collaboration among enterprises and individuals, International Standards (ISs) are supposed to be among the most accurate and unambiguous kinds of technical documents. Intentional ambiguities are characteristic of diplomatic agreements and sometimes legal documents, but they are scarce in ISs, because ISs aim to prevent or overcome incompatibilities that often arise when parties interact across national borders. Hence, ISs usually strive to specify methods and procedures unambiguously. However, ISs are often criticized as being difficult to use for reasons that include inter- and intra-standard inconsistency, low accessibility, poor traceability, and ambiguity.[1] The large number of authors and relationships to other domains renders managing the quality of technical documents in general and ISs in particular a daunting task,[2] calling for a formalized approach to authoring ISs. Motivated by this need to formalize ISs by basing them on formal models[1] through the process of model-based standards authoring (MBSA), we set out to examine and model ISO/IEC/IEEE 15288:2015 based on its Final Draft International Standard (FDIS),[3] hereinafter referred to as 15288.[*] This IS serves as a case in point to demonstrate the viability of the MBSA approach to dramatically improving the quality of ISs in terms of consistency and completeness.

One of the most established and cited ISs (e.g., refs. 4–7), 15288 is the basis for many key systems engineering processes. These include IEEE Std 15288.2™-2014 for technical reviews and audits during the US Department of Defense (DoD) and other defense agencies' acquisition life cycle[4] and IS 21840 – Guidelines for the utilization of 15288 in the context of system of systems (SoS) engineering.[5] The 15288 IS is also a basis for teaching systems engineering.[6] Together with two past standards, ANSI/EIA-632[7] and IEEE-1220,[8] 15288 plays a most important role as the source for processes in systems architecting, design,

[*] There is a 2023 version of 15288 which was not available and is not considered in this paper, because rather than criticizing 15288 as a main target, the objective of this research is to make the case for MBSA, using 15288 as a running example.

and integration.[9] In ref. 2, we have pointed out and demonstrated the value of basing standards on conceptual models.

Attempts to use conceptual modeling to enhance standards have been limited. Gomez et al.[10] proposed a methodology based on EIA-632[7] for systems engineering best practices using SysML as the system description language. In the context of semiconductor fabrication environment, capturing and transferring information throughout the product development cycle is critical. To solve the problem of information standards in this domain, which are incomplete, cumbersome, or unworkable, Simmon et al.[11] have proposed UML class diagram as a means to reduce redundant efforts of standards development teams, who can avoid revisiting due to missing or conflicting data and ensure that all of the necessary data are captured.

Anda et al.[12] reported empirical evaluation of UML-based development of large software projects by examining a case study in ABB in which a new version of a safety-critical process control system was developed to enable certification of embedded software according to the IEC 61508 safety standard. Interviewees experienced improvements with traceability, communication, and documentation, but the positive effects were reduced due to reasons including lack of training and inadequate modeling tools.

Orellana and Mandrick[13] have attempted to formalize 15288 using a systems engineering reference ontology based on Basic Formal Ontology (BFO)[14] in order to achieve interoperability and eliminate syntactic and semantic differences. They pointed out that 15288 needed to "be injected with a modeling mindset." The model would serve their motivation to find deficiencies and inconsistencies, correct them as necessary, and reduce the number of synonyms to prevent their misinterpretation by organizational judgement or bias. Ontologies are cited widely in ref. 13 as an approach to formalizing systems engineering, making systems interoperable, and enable time-based reasoning and consistent systems engineering measurement.

Standardizing Object-Process Methodology (OPM)[16,17] as an IS was initiated in by the International Organization for Standardization (ISO) following the 2008 INCOSE International Symposium in Utrecht, The Netherlands. The initial motivation for standardizing OPM by ISO was to have a standard modeling language and methodology that can serve as a basis for model-based standards authoring (MBSA). OPM was the first conceptual modeling language to become an ISO Publicly Available Specification (PAS) 19450[18] in 2015, and it is undergoing the process of becoming an IS by the end of 2023. The underlying OPM ontology is a highly compact universal top-level ontology. In OPM, the only two concepts are objects as things that exist physically or informatically, and processes, which are things that transform objects by creating or consuming them, or by changing their state. Due to its compactness and universality, OPM can serve to model any kind of phenomenon or system, as well as ISs, as modeling 15288 in this work demonstrates, helping to enhance interoperability and minimize syntactic and semantic differences, as aspired in.[13] The choice of OPM as the basis for MBSA is based on its universality (objects and processes), simplicity (only one kind of diagram), and bimodality (visual intuitive yet formal diagrams that are translated on the fly to OPL—a subset of English).

Realizing the immense value of formally modeling SE-related standards in general and 15288 in particular, in a previous work,[15] we have modeled the first material clause of 15288, Clause 5.2, titled *System concepts*, using OPM ISO 19450[18] to best reflect the implied authors' intent.

As Orellana and Mandrick[13] had expected, the 15288 modeling activity[15] brought up a number of significant ambiguities that arise from lack of coherence between the text and the figures, clearly pointing to the value of having a formal conceptual model as the underlying backbone of standards in general and this central standard in particular. Building on this pilot study, in this paper, we model the heart of 15288 – Clause 6, which describes system lifecycle processes.

While Clause 5 of 15288 is a prolog that focuses on systems and related concepts, Clause 6 is the central and longest part of 15288, where the details of the system lifecycle processes are described.

The system's function is the main system's process along with the operand (object) it transforms, which delivers value to the system's beneficiary. In the following section we demonstrate the methodological part of OPM by stepwise modeling of a common system so we can understand how to apply it to 15288.

While the focus of 15288 is on processes, applying OPM fosters and almost enforces parallel modeling of the objects—in our case, mostly informatical artifacts—associated with the processes. Hence, a major objective of the modeling is to highlight the objects—the artifacts involved in 15288—which are implicit to a large extent. This balanced emphasis on both processes—the procedural-dynamic aspect of the system lifecycle, and objects—its structural-static aspect, fosters a complete, holistic, formally well-defined, yet humanly accessible, representation of the two major system aspects, that is, structure and behavior, enabling the understanding of the third aspect—the system's function.

The formality of a language achieves the highest level of rigor through mathematical notation. OPM, which is bimodal, is a formal language. Each OPM model is expressed in two modalities—graphical and textual. The syntax of each modality is defined formally using a mathematical notation. The mathematical notations used for the graphical and textual modalities are graph grammar and Extended Backus-Naur Form (EBNF), which fully specifies its context free grammar conforming to ISO/IEC 14977:1996, respectively. The graphical modality of OPM is a hierarchically arranged set of Object-Process Diagrams (OPDs), which can be depicted manually or via an OPM modeling software, such as OPCAT or OPCloud.[19] The OPD set is fully specified via a graph grammar in,[20] enabling validation of existing OPDs for syntactic correctness and the creation of new OPDs that are syntactically correct. The textual modality of OPM, called Object-Process Language (OPL), is a constrained subset of English (and other natural languages) that is generated automatically by the OPM modeling software. Like a programming language, OPL is formal, as its syntax is defined completely and unambiguously in Annex A of ISO 19450:2015.[18]

OPM was selected as the basis for the model-based standards authoring (MBSA) approach due to its formality and graphics-text bimodality, which does not exist in SysML.[17] OPM is simpler than SysML as it has just one kind of diagram, OPD, as opposed to the

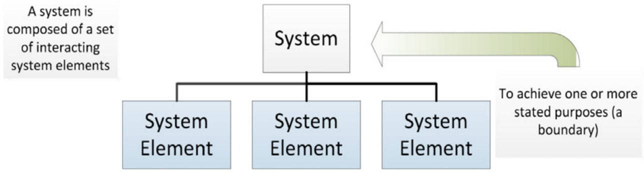

FIGURE 1 Figure 1 of 15288, titled "System and system element relationship".

nine kinds in SysML. OPM is also bimodal: Its diagrams are automatically translated into simple sentences in OPL, a constrained subset of English. OPM's simplicity and bimodality makes it very fit as the MBSA foundation, as this work demonstrates.

2 | SYSTEM STRUCTURE ACCORDING TO ISO 15288

Before delving into the modeling of system lifecycle processes in Clause 6 of 15288, which is the heart of 15288, we recap the OPM modeling of Clause 5.2 – *System concepts* done in.[15] Clause 5.2.2, *System Structure*, contains Figure 1 of 15288, shown here in Figure 1 [†], and reference to it in the text, which reads: "*The system life cycle processes in this International Standard are described in relation to a system (see Figure 1) that is composed of a set of interacting system elements, each of which can be implemented to fulfill its respective specified requirements*."

Figure 1 of 15288 contains two blocks of text that do not sit well with the main text cited above. The first part of this text, "*A system is composed of a set of interacting system element*" is repeated unnecessarily in the top-left textbox in Figure 1, but the figure does not include any graphical representation of this interaction. In the main text, the second part of this sentence is: "*… each of which can be implemented to fulfill its respective specified requirements*". The analogous text in the bottom-right box of Figure 1 reads: "*To achieve one or more stated purposes (a boundary)*". These two parts of the same sentence, the one in the main text and the other in Figure 1, are different: The main text relates to the fact that each system element *can be implemented to fulfill its respective* ***specified requirements***, while the text in the figure talks about the need to achieve *one or more* ***stated purposes***, implying that ***purposes*** and ***requirements*** are equivalent.

Significant efforts by many experts were expended in creating 15288. The misalignment between the semantics expressed by the text on the one hand and the figures on the other, as demonstrated above, may be a result of insufficient verification of the correspondence between these two complementary modalities. A similar semantic discrepancy between text and graphics in international standards was also demonstrated in[2] for the Personnel Model in ISO/IEC 62264, indicating that differences between text and graphics is likely to be a widespread phenomenon that needs attention. Additional problems of this nature are elaborated on in ref. 21.

† "Fig. *i*" refers to the figure number *i* in this paper, while "Figure *j*" refers to figure number *j* in 15288.

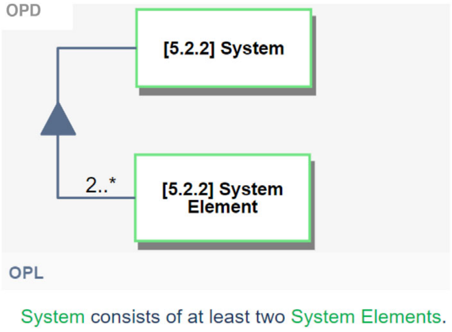

FIGURE 2 The graphical part of Figure 1 as an OPM model (Note: OPL is designed to omit brackets and the content they enclose).

The problem of mismatch between the main standard text and the accompanying figures is widespread across many ISs and their drafts in various preparation stages. While preparing the draft of ISO 17649, titled "Model-Based Standards Authoring,"[22] the major comment was lack of examples to justify the MBSA effort. In response, the following version of that draft was enriched with comprehensive examples, including the following: (1) 15288, (2) IEC 62264–5_2016 – Enterprise-Control System Integration—Transaction Models, (3) Process Reference Model (PRM) ISO/IEC/WD1/42020-2015, (4) Simulation interoperability reference model and process, (5) Environmental performance evaluation (EPE) data aggregation process ISO 20140–3:2016(E), (6) ISO/TC184/SC5/WG16 – Supply chain interoperability and integration, (7) OpenSCENARIO by the Association for Standardization of Automation and Measuring Systems (ASAM), and (8) Resource Management Services (RMS) ISO 20242-2(E):2010. The latter, for example, specifies that the RMS interface may support any number of different types of peripheral interfaces which may have any number of communication channels. Figure 4 in that document, shows this with an example of two different regions and sub-regions that are identical. The legend of that figure reads "multiple peripheral interfaces," while the figure contains exactly two interfaces without specifying "peripheral", but instead specifying "InterfaceTypeSelected". This example demonstrates the need for a higher level of formality and correspondence between the figure and the text that is supposed to explain it.

The confusing misalignment between the main text of 15288 and the accompanying figures, which are supposed to enhance its understanding, calls for applying a formal modeling language to underpin the messages that this IS aims to convey. Hence, in the next section, we model 15288 gradually. The OPM-based modeling has two objectives: (1) refining and crystalizing our understanding of 15288 as a central IS, and (2) demonstrating the value and viability of using a formal conceptual model as the underlying backbone of any technical document in general and ISs in particular. This activity is in line with the Terms of Reference for Working Group 15 – Model-Based Standards Authoring of ISO TC 184/SC 5,[23] which calls for "*developing a model-based methodology of authoring new standards and revision of existing ones based on the use of ISO PAS 19450:2015 OPM modeling language and methodology.*"

Figure 2 is an OPM model of 15288 Figure 1 (Figure 1 also here). In Figure 2, the OPD is the top part of the model and the OPL—the

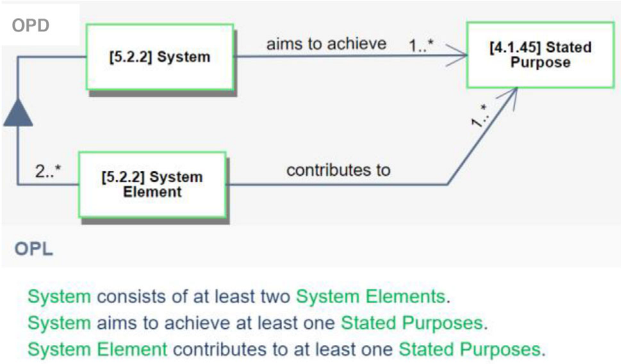

FIGURE 3 **Stated Purpose** from Figure 1 added and connected to **System** and **System Element** from the model of Figure 2.

bottom. This unique bimodal representation of OPM—the graphical and the textual—makes it especially suitable for model-based standards authoring (MBSA),[15,23] because the text, which is generated automatically, is a faithful reflection of the formal verifiable graphical representation of the system which the IS focuses on.

The numbers in brackets, such as **[5.2.2]**, are the 15288 clause or subclause numbers in which the concept that follows the brackets is defined or explained for the first time. Differences between Figure 2 and Figure 1 include (1) Unlike Figure 1, in Figure 2, **System Element** appears graphically as an object only once, with the multiplicity notation 2..* next to **System Element**, expressed by the OPL sentence "**System** consists of two to many **System Elements**." (2) OPM enables distinction between physical things (objects and processes), which are shaded, and informatical things, which are not shaded. In OPM, the **Essence** attribute of a **Thing** can have one of two values: **physical** or **informatical**. Obviously, 15288 applies to both physical and informatical systems. However, even a purely informatical system is ultimately carried out by a physical medium. Since 15288 does not specify the essence of neither the system nor the system elements, the OPD in Figure 2 arbitrarily expresses both **System** and **System Element** as being physical. (3) The link with the black triangle pointing to **System** symbolizes the OPM aggregation-participation structural relation. (4) The OPL sentences express exactly what the graphical modality, OPD, expresses, so each modality reinforces the understanding of the other, catering to humans' dual channel processing.[24]

To better align with 15288, in Figure 3 we added the object **Stated Purpose** to the model of Figure 2 and connected it to both **System** and **System Element** from the model of Figure 2 using tagged structural links. These additional OPL sentences convey the intended meaning. **Stated Purpose** is an informatical object, so it is not shaded.

Next in 15288 comes Figure 2, presented here as Figure 4, along with the following explanation (Subclause 5.2.2): "For more complex systems-of-interest, a prospective system element may itself need to be considered as a system … before a complete set of system elements can be defined with confidence." In short, a system element may be a system. The OPM model in Figure 5 is a formal representation that specifies this more accurately, making Figure 2 of 15288 more general and an order of magnitude more compact than Figure 4.

Subclause 5.2.3, Enabling systems, contains the following text: "In addition to interacting with enabling systems, the system-of-interest may also interact with other systems in the operating environment, shown as Systems A, B, and C. Requirements for interfaces with enabling systems and other systems in the operational environment will need to be included in the requirements for the system-of-interest." What is really surprising is that in Figures 2 and 1 of 15288, drawn in this IS just one and two pages earlier, respectively, **boxes** were used to symbolize systems and their variants, while in Figure 3, **ellipses** serve to denote the same concept. Why graphical symbols expressing the same concept change from one diagram to the next is unclear.

The multiplicities in the OPM model in Figure 5 are based on the main text or deduced from the examples provided in 15288 Figure 2 (Figure 5 here). Clause 5.5.2 of 15288 states that "*The system life cycle processes … are described in relation to a system (see Figure 1) that is composed of a set of interacting system elements…*". This implies that a system must be composed of at least two system elements. Inspecting 15288 Figure 2 (Figure 5 here) reinforces this statement. It reveals that white boxes represent **System** (and **System-of-Interest** as a specialization of **System**), while blue boxes represent **System Elements**. There are eight white boxes—entities of type **System**, each comprising an optional **System** and at least two **System Elements**. This leads to the assumption that the minimal number of **System Elements** in any **System** is 2, which is supported by the conclusion that if a **System** consists of just one **System Element**, the **System Element** and the **System** are one and the same, making **System Element** redundant. This constraint is expressed in the OPL sentence as "**System** consists of at least two **System Elements**." This is the OPL interpretation of the multiplicity "2..*" from **System** to **System Element** in the OPD in Figure 5. The OPL sentence "**System Element** may be a **System**." is based on the fact that each of the white boxes in 15288 Figure 2, which represent **System**, covers a blue box, which represents a **System Element**, implying that a **System Element** in one **System** can be another **System** in its own right.

Figure 3 in 15288, depicted here as Figure 6, uses ***Operational*** *Environment*, but the main 15288 text relates to ***operating*** *environment*. It is not clear why both terms, ***operating*** *environment* and ***operational*** *environment*, are used, and one cannot tell whether these concepts are the same or not.

Seeking an explanation to 15288 Figure 3, we find the following text: "The relationship between the services delivered to the operational environment by the system-of-interest and the services delivered by the enabling systems to the system-of-interest are shown in Figure 3." Yet, in Figure 3 the word services is missing. Instead, we find Interaction. If one assumes that the figures and the text are aligned, the inescapable conclusion is that services and Interaction are the same. Obviously, however, these are two distinctly different concepts: While a service requires interaction between two or more parties such that the server—the serving party—provides value to the party being served, not every interaction is a service, because not every interaction provides value to any one of the interacting parties.

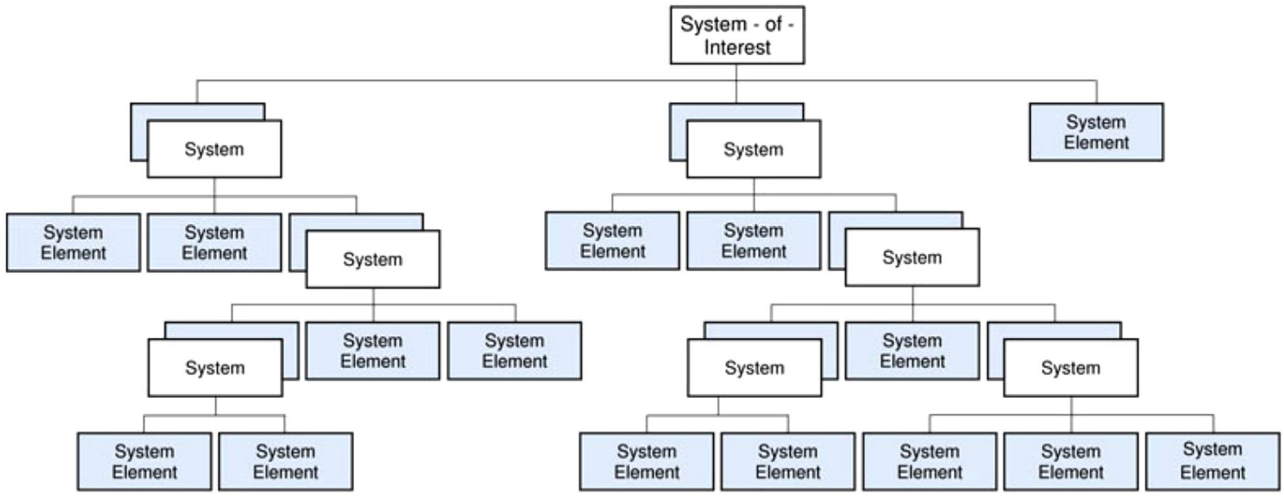

FIGURE 4 15288 Figure 2: The structure of System-of-Interest.

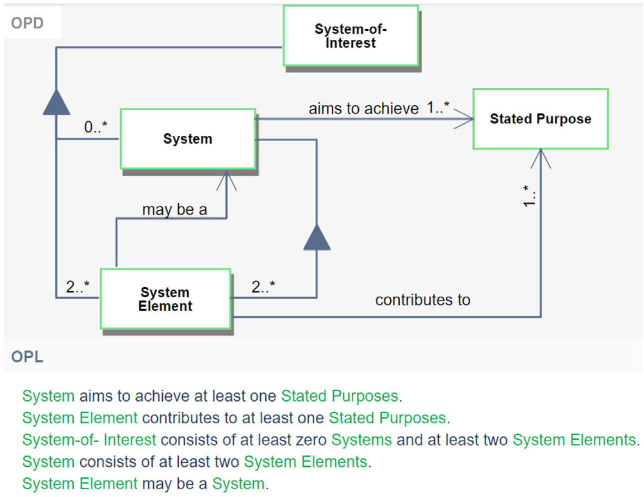

FIGURE 5 OPM model of System-of-Interest depicted in 15288 Figure 2.

Subclause 5.2.3 specifies that "*The interrelationships between the system-of-interest and the enabling systems can be bi-directional or a one-way relationship.*" To conform with this text, some of the arrows in 15288 Figure 3 should be bidirectional and the others need to be uni-directional, some from a system-of-interest to an enabling system and others in the reverse direction.

Further, in 15288 Figure 3, a distinction is made between **Systems in Operational Environment** and **Enabling Systems**, by both name and hatching. But the difference between these two kinds of systems is unclear. In Subclause 4.1.7, **enabling system** is defined as a *system that supports a system-of-interest during its life cycle stages but does not necessarily contribute directly to its function during operation*. Examining 15288 Figure 3, one might conclude that the set comprising **Systems in Operational Environment** and that of **Enabling Systems** are disjoint. There is, however, a possibility that an enabling system is in the operational environment, but it is not considered in the text of 15288. Additional puzzling, confusing findings are elaborated on in.[15]

The recurrent inconsistencies between the text and the first three figures further strengthen the call for adopting a higher level of formality via conceptual modeling of 15288. This approach eliminates various kinds of inconsistencies, including internal text consistency, text versus figure consistency, and inter-figure consistency, as elaborated in.[2] OPM is suitable for this purpose, as its formality is anchored in ISO 19450, and it uniquely generates natural language which is an accurate reflection of the formal graphical representation.

3 | FROM OBJECTS TO PROCESSES: THE DISCONNECT BETWEEN 15288 CLAUSES 5 AND 6

Figure 4 in 15288 – System life cycle processes, provided here as Figure 7, looks like a table of content for Clause 6, with references to the subclauses and classification of the processes into four process groups: Agreement, Organizational Project-Enabling, Technical Management, and Technical Processes. From an OPM viewpoint, a process is a thing that transforms (creates, affects, or consumes) at least one object, but here, the objects which are transformed by these processes are not explicitly discussed. As objects are the informatical or physical artifacts involved in any system lifecycle, an International Standard cannot afford to almost ignore them. This lack of reference in Clause 6—the heart of 15288—to objects is thus sorely missing. Moreover, there is a sharp disconnect between Clause 5, which is object-oriented, and Clause 6, which is process-oriented, making it seem that these two clauses were written by different groups of people, with little effort to integrate them.

4 | TWO 15288 TASK DEFINITIONS?

In Clause 4.1.49, a task is defined as "*required, recommended, or permissible action, intended to contribute to the achievement of one or more outcomes of a process.*" This definition is modelled faithfully in Figure 8,

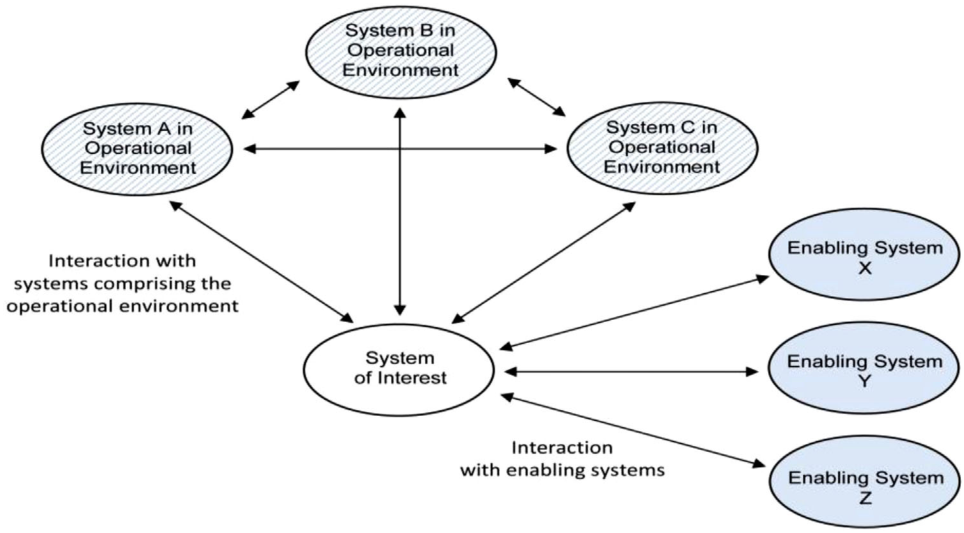

FIGURE 6 15288 Figure 3: System-of-interest and related systems.

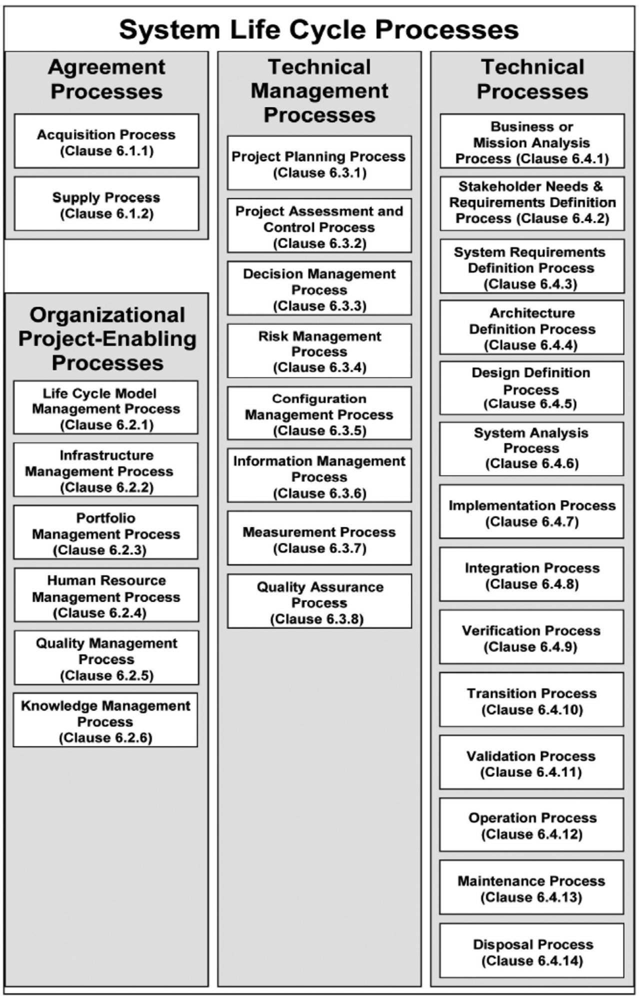

FIGURE 7 System life cycle processes, depicted in 15288 as Figure 4.

as expressed by the OPL. Yet, in 5.5.2 we read: "*The tasks are requirements, recommendations, or permissible actions intended to support the achievement of the outcomes.*" While superficial reading may not reveal the principal difference between this task definition and the previous one, the OPM model in Figure 9 clarifies that in addition to being a **Permissible Action**, a **Task** can also be a **Requirement** and a **Recommendation**, contradicting Clause 4.1.49, according to which a task is an action (required, recommended, or permissible), but neither a requirement nor a recommendation! Obviously, a reader of 15288 must not face the dilemma of what definition should be considered the "correct" one, especially given the centrality of the task concept in this IS. However, in order to proceed, we must make a choice, and we elect to go with the first one, given in 4.1.49.

The definition of **process** in 15288 Clause 4.1.29 is "*set of interrelated or interacting **activities** that transforms inputs into outputs*". The definition of **activity** in 4.1.3 is "*set of cohesive **tasks** of a process*". Finally, the definition of **task** in 4.1.49 reads: "*required, recommended, or permissible **action**, intended to contribute to the achievement of one or more outcomes of a **process***". Hence, task is the lowest in this three-level process-activity-task hierarchy, which implies that it is atomic.

5 | THE 15288 PROCESS-ACTIVITY-TASK HIERARCHY

According to 15288 Clause 5.6.1, there are four process groups. Adding **Process Group** as a level above **Process** yields a four-level hierarchy, depicted in Figure 10a: **Process Group**, **Process**, **Activity**, and **Task**. However, the definition of task does not preclude it from being complex enough, requiring decomposition into lower level subprocesses, which, in turn, may require yet another level of decomposition, and so on. Indeed, 15288 contains process specifications that require more than four levels of depth, mandating that the hierarchy must not be restricted to four levels. Consider, for example,

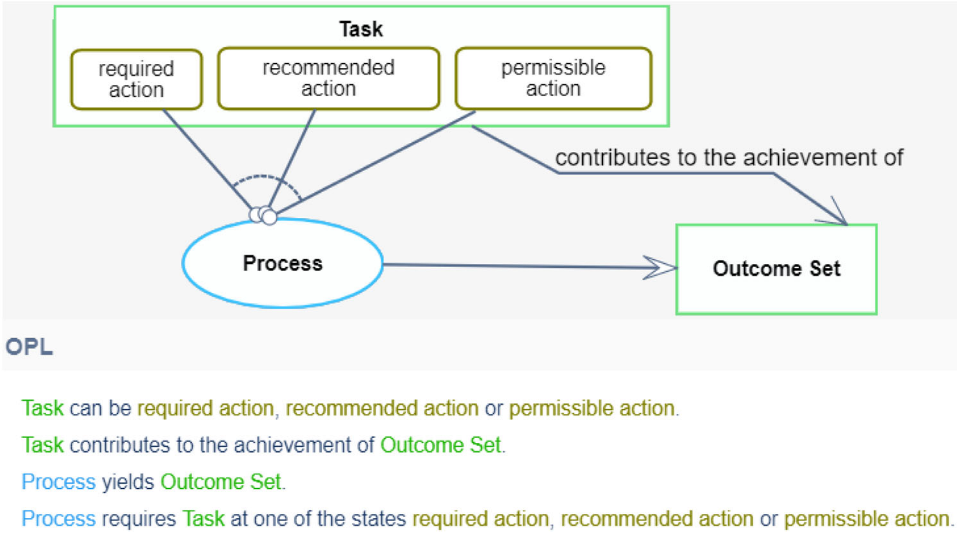

FIGURE 8 OPM model of **Task** according to 15288 Clause 4.1.49.

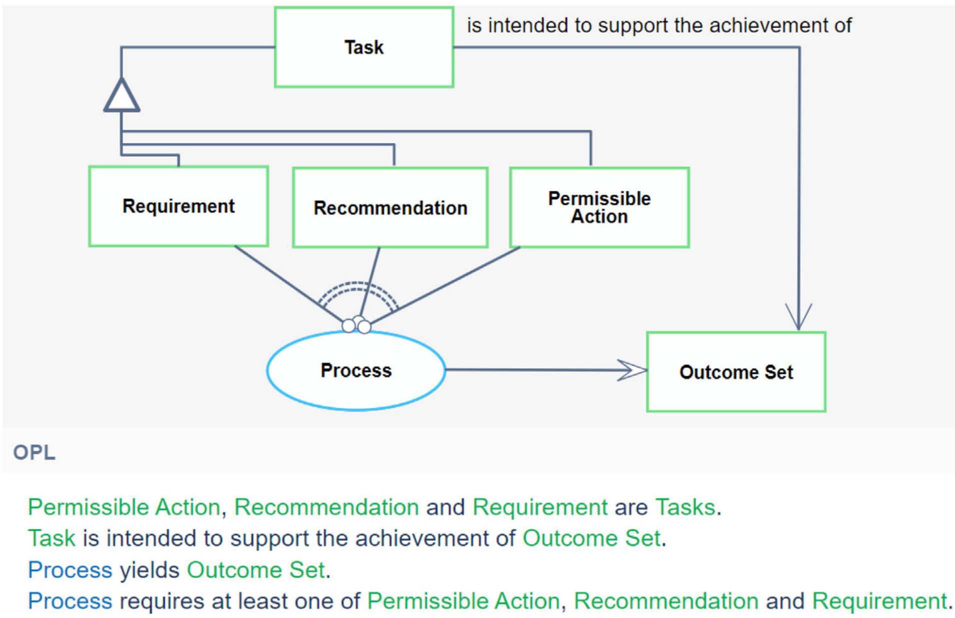

FIGURE 9 OPM model of **Task** according to 15288 Clause 5.5.2.

System Life Cycling in Figure 11 [Clause 6] as the top level {level 1} and **Agreement Process Set** [Clause 6.1] in Figure 12 as it subprocess {level 2}. In Figure 14, **Agreement Process Set** is shown to comprise **Acquisition** [Clause 6.1.1] and **Supply** [Clause 6.1.2]. Continuing, in Figure 15, **Acquisition** includes **Acquisition Preparing** [Clause 6.1.1.3a] {level 3}and four additional subprocesses. One of the subprocesses of **Acquisition Preparing**, listed in Clause 6.1.1.3a) 2), is **Request For Supply Preparing** {level 4}, which is not included as an OPD due to lack of space. Further, **Request For Supply Preparing** consists of **Requirements Developing** {level 5} and **Requirements Approving** {level 5}. Without further decomposing any one of the last two subprocesses we already have a five-level hierarchy. The 15288 IS does not claim completeness for system process descriptions, and the notion of process decomposition into four successive extents of granularity is constrained for the purpose of identifying labels for subordinate and dependent actions in the text. OPM solves the limited hierarchy problem as modelled in Figure 10b, avoiding the use of different names for the various process levels and referring to all as processes. Each **Process** is comprised of two or more **Subprocesses**. **Atomic Process** is a specialization of **Subprocess** which does not consist of any lower-level **Process**, allowing for any number of process levels.

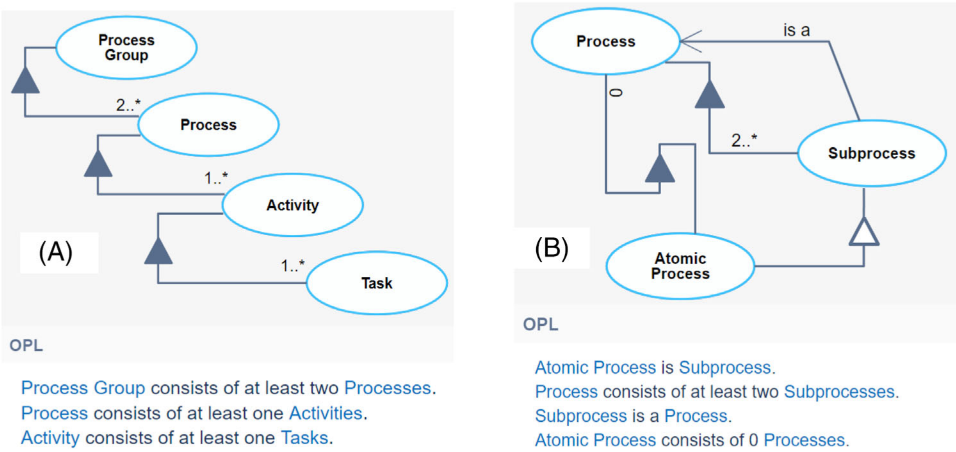

FIGURE 10 Process hierarchy OPM models: (A) The four-level hierarchy according to 15288. (B) A generic, unlimited hierarchy.

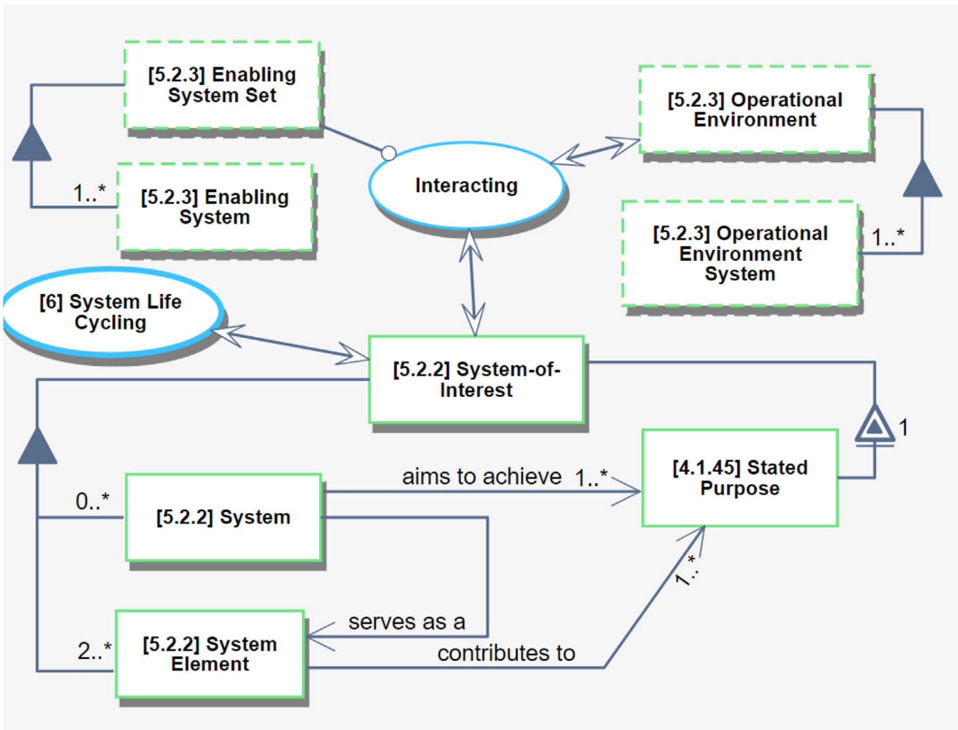

FIGURE 11 OPM model of System-of-Interest and the effect of System Life Cycle Process Set on it.

6 | OVERCOMING THE OBJECT-PROCESS DISCONNECT

We proceed with Figure 11, which is an OPM model of the **System-of-Interest** in Figure 5 with the addition of what 15288 Figure 3 expresses informally and the main process, **System Life Cycling**, as the missing link between Clause 5 and Clause 6. The effect (bidirectional arrow) is translated to the OPL sentence **System Life Cycling** affects **System-of-Interest**. We also added the process connected it to **Enabling System Set** with an instrument link, indicating that this object is an instrument for the **Interacting** process. The effect link (bidirectional arrow) between **Interacting** and **System-of-Interest** implies that the interaction may occur also between its parts, which are **System** and **System Element**. The resulting OPM model demonstrates both the level of formality and the human accessibility: Every statement is represented unambiguously in both graphics and text, capitalizing on humans' dual channel processing.[19]

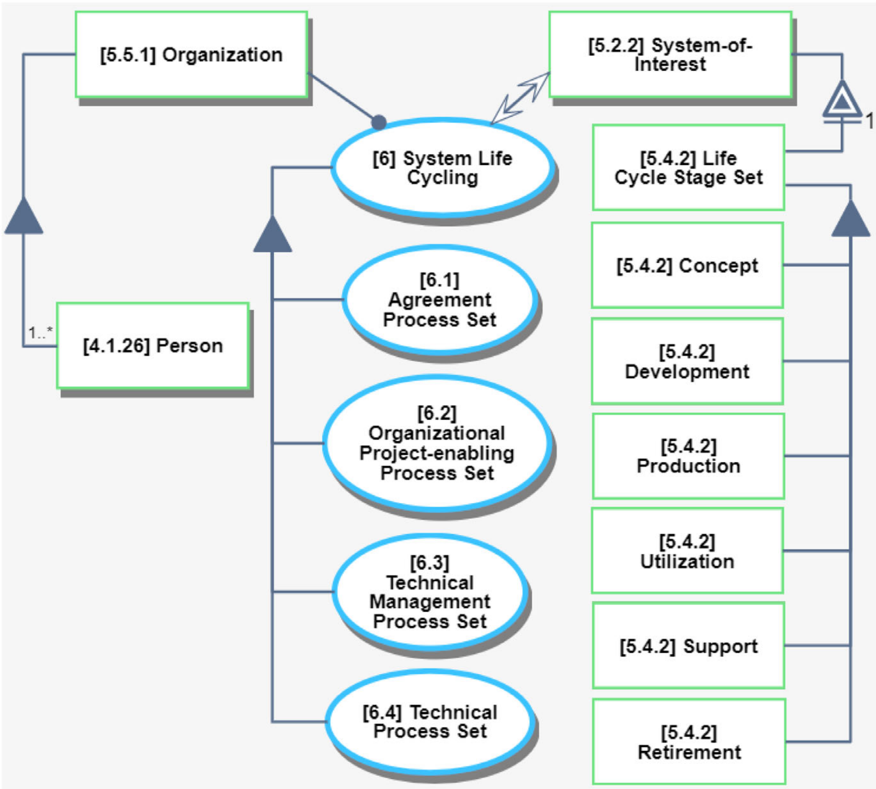

FIGURE 12 Unfolding **System Life Cycling** and **Life Cycle Stage Set** reveal the four process sets [6.1-6.4] and the six life cycle stages [5.4.2].

7 | MODELING THE FOUR PROCESS GROUPS

OPM has two main process refinement mechanisms: (1) unfolding, used for asynchronous processes—those having no predefined order, and (2) in-zooming, used for synchronous processes—those having a predefined order. In Figure 12, **System Life Cycling** is unfolded, revealing its four process sets [6.1 through 6.4]. The four process groups are unfolded because they are asynchronous. For example, technical management processes and technical processes can indeed happen simultaneously or be intertwined in any order. **Life Cycle Stage Set**, an attribute of **System-of-Interest**, is also unfolded, exposing the six life cycle stages in Clause 5.4.2.

In Figure 13, selected OPL sentences of the OPD in Figure 12 are presented, clarifying the semantics of the graphic symbols and providing a living textual documentation that is in line with the graphics at all times. For example, the OPL sentence "**System-of-Interest** exhibits **Life Cycle Stage Set**." expresses the fact that **Life Cycle Stage Set** is an attribute of **System-of-Interest**, which is expressed graphically by the link with the black-in-white triangle at the top-right corner of Figure 12. Incidentally, a matrix in 15288 showing what process groups are involved in what stages could be beneficial.

System Life Cycling consists of Agreement Process Set, Organizational Project-enabling Process Set, Technical Management Process Set, and Technical Process Set.
System-of- Interest exhibits Life Cycle Stage Set.
Life Cycle Stage Set consists of Concept, Development, Production, Retirement, Support, and Utilization.
Organization consists of 1 to many People.
System Life Cycling is physical.
Organization handles System Life Cycling.
System Life Cycling affects System-of- Interest.

FIGURE 13 Selected OPL sentences from the OPL paragraph corresponding to the OPD in Figure 12.

8 | ZOOMING INTO THE ACQUISITION PROCESS

Following the depth-first traversal strategy that led the authoring of 15288, we continue by drilling down into Agreement Processes. Based on the text below, Figure 14 is the OPD of the **Acquisition** process (the first part of **Agreement Process Set**), its **Purpose** [6.1.1.1], and the **Acquisition Outcome Set** [6.1.1.2]. Clause 6.1.1.1 reads as follows.

"*6.1.1 Acquisition process – 6.1.1.1 Purpose*"

"*The purpose of the Acquisition process is to obtain a product or service in accordance with the acquirer's requirements.*"

For example, the value of the attribute **Purpose** [6.1.1.1] of **Acquisition** [6.1.1], recorded in Figure 14 bottom right, is "**to obtain a product or service in accordance with the acquirer's requirements**", yielding an OPL sentence which states exactly this. As we continue reading, in

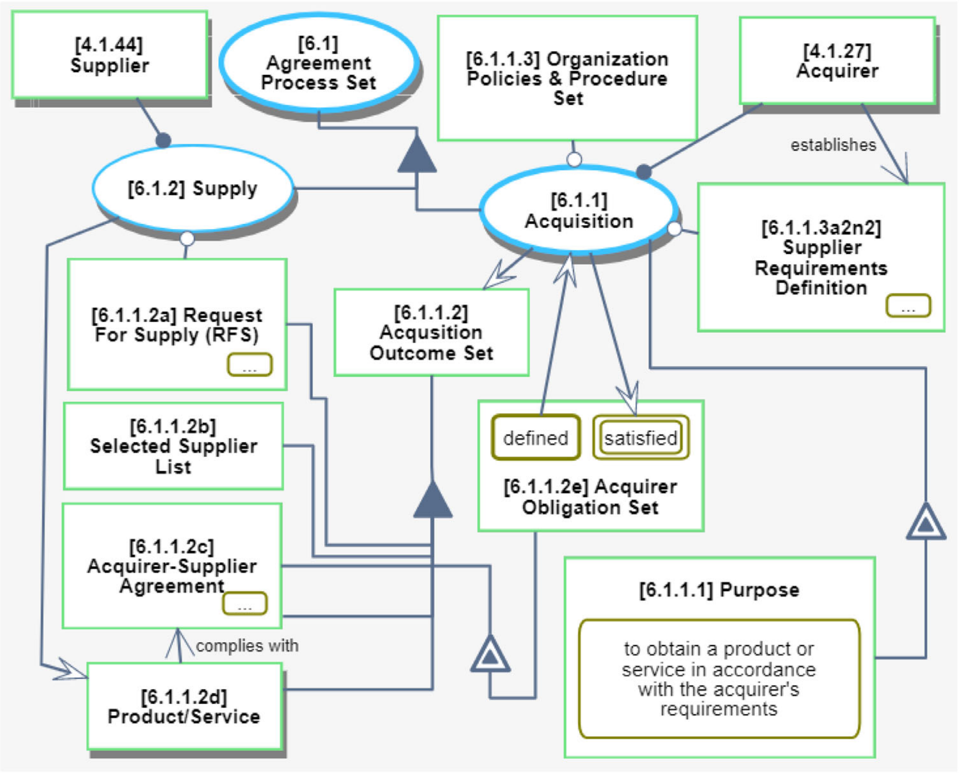

FIGURE 14 OPD of **Acquisition** as part of **Agreement Process Set**, its **Purpose** [6.1.1.1] and **Acquisition Outcome Set** [6.1.1.2].

6.1.1.2 we get the **Acquisition Outcome Set**: "*As a result of the successful implementation of Acquisition process: a) A request for supply is prepared. b) One or more suppliers are selected. c) An agreement is established between the acquirer and supplier. d) A product or service complying with the agreement is accepted. e) Acquirer obligations defined in the agreement are satisfied*." Outcomes a-d are the objects 6.1.1.2a-6.1.1.2d at the bottom left of Figure 14, the last one being the physical **Product/Service**, which results from the process **Supply** [6.1.2] and "**complies with**" **Acquirer-Supplier Agreement** [6.1.1.2c], in accordance with d) above. Item e) is modelled as the object **Acquirer Obligation Set** [6.1.1.2e], whose states change from **defined** to **satisfied** by the **Acquisition** process. **Acquirer** is defined in 4.1.1 as "*stakeholder that acquires or procures a product or service from a supplier*," while **Supplier** is defined in 4.1.44 as "*organization or an individual that enters into an agreement with the acquirer for the supply of a product or service*." Why the former is a "*stakeholder*" while the latter is "*organization or an individual*" is unclear, as one would expect them to be symmetrical. Stakeholder is not defined in Clause 4, but in Clause 5.3.1 we indeed find that "'*stakeholder' refers to an individual or organization with an interest in the system*."

9 | ZOOMING FURTHER INTO ACQUISITION

Acquisition is refined to a fourth detail level in Figure 15. Here, the refinement applies in-zooming, because the process is synchronous—its subprocesses follow a well-defined order, depicted by their top-to-bottom arrangement. rather than unfolding. In contrast, the refinement in Figure 12 was done by applying unfolding, because the four process sets (which are the subprocesses of **System Life Cycling**) are asynchronous.

Clause 6.1.1.3 – *Activities and tasks*, lists the following five main subprocesses of **Acquisition** (the corresponding process name in the OPD in Figure 15 follows): *a) Prepare for the acquisition*—**Acquisition Preparing**; *b) Advertise the acquisition and select the supplier*—**Acquisition Advertising & Supplier Set Selecting**; c) *Establish and maintain an agreement*—**Agreement Establishing & Maintaining**; *d) Monitor the agreement*—**Agreement Monitoring**; and *e) Accept the product or service*—**Product/Service Accepting**.

10 | SUMMARY, CONCLUSION, AND FUTURE WORK

This paper demonstrates the model-based standard authoring (MBSA) approach to authoring new standards and revising existing standards. As a running example, 15288 was selected due to its prominence and centrality to the systems engineering community. The effort to author 15288 has been gigantic, resulting from contributions of many individuals over many years. Critique of 15288 standard should by no means be interpreted as its assessment of not being at a high quality. The 15288 standard is largely accurate, and a standard does not need to be perfectly correct and complete to be useful. The objective of the paper is not to criticize 15288 or show its deficiencies, but to demonstrate the value of the MBSA approach and its potential to improve not only

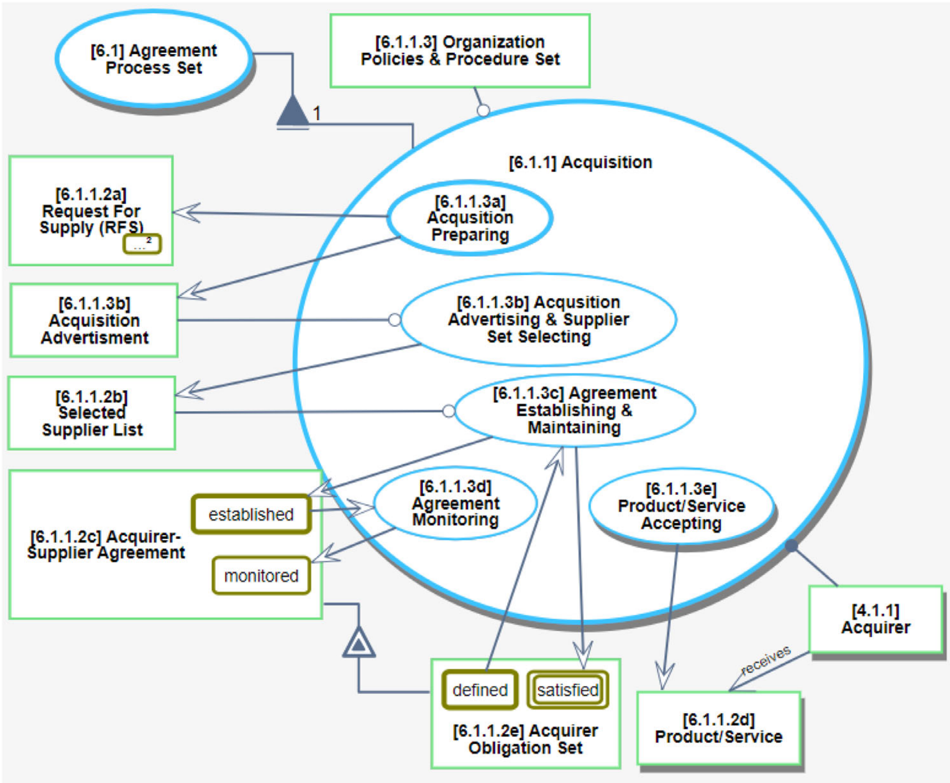

FIGURE 15 OPD of **Acquisition** in-zoomed, exposing the five subprocesses and their outcomes, as specified in 15288 Clause 6.1.1.3.

15288 but any current or future international standard. An overly formal standard might make it less useful to its intended audience. Yet, a formal modeling approach can be useful, for example, to detect objects that are missing as inputs or outputs to processes, processes that transform objects to other objects, and the use of more than one word or phrase to refer to the same concept, which may cause confusion and uncertainty regarding the intent of the text.

The model covers the specifics of 15288 at all detail levels till subsection 6.1.1.3. More work is needed to complete the model, but the current model provides proof-of-concept of the value of MBSA. The resulting model can be used as a basis for a systems engineering ontology, which not only lists concepts and relations, but also presents formally, clearly, and unambiguously the involved organizations, humans, their tasks for each process, and the required inputs and resulting artifacts (objects) from each process.

ISO TC184/SC5/WG15[23] has developed a draft of ISO 17649 titled “Model-Based Standards Authoring.”[22] The value of the MBSA approach has been demonstrated in this draft by converting to OPM models informal or various semi-formal diagrams or specifications found in current international standards or drafts thereof, including 15288. Other modeled examples include the Interoperability Reference Model & Process (SIRMoP) framework, Resource Management Services (RMS) ISO 20242-2(E) 2010, Process Reference Model (PRM) ISO/IEC/WD1/42020-2015, Environmental performance evaluation (EPE) data aggregation process ISO 20140–3:2016(E), IEC 62264–5 2016(E F)—Enterprise-Control System Integration—Transaction Models, Constructs for Enterprise Modeling, OpenSCENARIO by ASAM—Association for Standardization of Automation and Measuring Systems, and Digital Engineering Measurement Framework Version 0.95, 2022.

The contribution of this paper is at two levels: At the immediate level, the complete model of 15288 is expected to raise the conciseness and usability of this key International Standard. Conciseness is defined as “the quality of being short and clear, and expressing what needs to be said without unnecessary words.[25]” In the context of this paper, conciseness is given a comprehensive interpretation, which, in addition to terseness and clarity, also includes precision, completeness, and consistency (lack of contradictions). Since 15288 is widely referenced and applied throughout the Systems Engineering community, increasing these attributes is especially important. While the current 15288 focuses on system lifecycle processes, the OPM model provides a missing balance between the processes and the objects—the artifacts that are inputs to or output of these processes. At the more general and long-term level, this paper provides a proof of concept to the model-based standards authoring (MBSA) approach, which calls for basing international standards on a formal model, from which the text is derived.

The model underlying the IS has several benefits: (1) It raises the IS formality level, aligning the diagrams with the text and significantly reducing its internal inconsistencies. (2) It enables the representation of the IS in both graphics and text—two complementary modalities that cater to humans' dual-channel processing.[24] (3) It provides for verifiability via systematic comparison and cross-validation of closely related standards, enabling a common ontology. This, in turn, boosts users' confidence in ISs, consequently elevating their usability and increasing their adoption.

Future research aims to (1) use deep learning to make the text generated automatically from the OPM graphical part of the model more humanly pleasing to read while maintaining its faithfulness to the formally verified graphical part of the model, (2) automatically convert natural language text in ISs into OPM models, and (3) enhance the OPM simulation capabilities to enable testing the standard's underlying model.

DATA AVAILABILITY STATEMENT

The entire OPM model for this publication, including the OPDs and their respective OPL paragraphs, is accessible in this link. https://technionmailmy.sharepoint.com/:b:/g/personal/dori_technion_ac_il/EVNbBlez3XpNkPQZ2xgugvQB53XjTjSYlzcbMY_0HkO6Hg?e=x2QeN6

ORCID

Dov Dori https://orcid.org/0000-0002-2393-3124

REFERENCES

1. Dori D, Blekhman A, Martin R, Model-based standards authoring. In: Proc. 21st INCOSE International Symposium. Denver, CO, USA, 2011. Accessed January 13, 2021. http://esml.iem.technion.ac.il/wp-content/uploads/2011/07/Model-Based-Standards-Authoring-March-2011.pdf
2. Blekhman A, Wachs JP, Dori D. Model-based system specification with tesperanto: readable text from formal graphics. *IEEE SMC: Syst.* 2015;45(11):1448-1458.
3. ISO/IEC/IEEE, ISO/IEC/IEEE FDIS 15288:2014 - Systems and software engineering — System life cycle processes; 2014.
4. Roedler G, Davis D. *IEEE Standard for Technical Reviews and Audits on Defense Programs*. IEEE Std 15288.2™-2014, IEEE Computer Scoiety; 2015.
5. SO/IEC/IEEE 21840:2019 – Systems and software engineering—Guidelines for the utilization of ISO/IEC/IEEE 15288 in the context of system of systems (SoS)
6. Bougaa M, Bornhofen S, O'Connor RV, Riviere A, A standard based adaptive path to teach systems engineering: 15288 and 29110 standards use cases. 2017 *Annual IEEE International Systems Conference (SysCon)*. Montreal, QC. 2017, pp. 1-8.
7. Processes for Engineering a System—EIA632, 2003. Accessed January 14, 2021. https://www.sae.org/standards/content/eia632/
8. IEEE 1220–2005 - IEEE Standard for Application and Management of the Systems Engineering Process. 2005. Accessed January 14, 2021: https://standards.ieee.org/standard/1220-2005.html
9. Sage AP, Biemer SM. Processes for system family architecting, design, and integration. *IEEE Syst J*. 2007;1(1):5-16. doi:10.1109/JSYST.2007.900240
10. Gomez C, Esteban P, Pascal JC, Jimenez F. Production systems and supply chain management in emerging countries: best practices. In:G, Velasco. *From Embedded Systems Requirements to Physical Representation: A Model-Based Methodology in Accordance with the EIA-632 Standard*. Springer; 2012. doi:10.1007/978-3-642-26004-9_11
11. Simmon ED, Messina JV, Griesser A, Simple UML modeling to improve the development of information standards. AEC/APC Symposium XVII, Indian Wells, CA, USA, 2005. Accessed January 13, 2021. https://www.nist.gov/publications/simple-uml-modeling-improve-development-information-standards
12. Anda B, Hansen K, Gullesen I, Thorsen HK. Experiences from introducing UML-based development in a large safety-critical project. *Empir Softw Eng*. 2006;11:555-581. Accessed January 13, 2020. https://link.springer.com/article/10.1007/s10664-006-9020-6
13. Orellana D, Mandrick W. The ontology of systems engineering: towards a computational digital engineering semantic framework. *Proc Comput Sci*. 2019;153:268-276.
14. Arp R, Smith B, Spear AD. *Building Ontologies with Basic Formal Ontology*. The MIT Press; 2015: p. xvi.
15. Dori D. How can OPM-based modeling disambiguate system concepts in ISO/IEC/IEEE 15288?. *Proc. 14th Annual IEEE International Systems Conference (SysCon2020)*; 2020.
16. Dori D. *Object-Process Methodology—A Holistic Systems Paradigm*. Springer Verlag; 2002.
17. Dori D. *Model-Based Systems Engineering with OPM and SysML*. Springer; 2016.
18. ISO. ISO/PAS 19450 Automation systems and integration—Object-Process Methodology; 2015. https://www.iso.org/obp/ui/#iso:std:iso:pas:19450:ed-1:v1:en
19. Dori D, Kohen H, Jbara A. OPCloud: an OPM integrated conceptual-executable modeling environment for Industry 4.0. In: Kenneth R. Zoneneshain A, Swarz RS, eds. *Systems Engineering in the Fourth Industrial Revolution*. Wiley; 2020.
20. Bibliowicz A, Dori D. A graph grammar-based formal validation of object-process diagrams. *Softw Syst Model*. 2012;11(2):287-302.
21. Dori D, Martin R, Blekhman A, Model-based meta-standardization: modeling enterprise standards with OPM. 2010 IEEE International Systems Conference, San Diego, CA, USAl 2010.
22. ISO/AWI 17649 Model-Based Standards Authoring. ISO TC 184/SC 5, June 2023.
23. ISO/TC 184/SC 5 N1826 "*Interoperability, integration, and architectures for enterprise systems and automation applications—Terms of Reference for WG15*. ANSI (United States), 2020.
24. Mayer RE. *Mulitmedia Learning*. 2nd ed. Cambridge University Press; 2011. ISBN10 0521735351.
25. Cambridge Dictionary. 2021. Accessed January 13, 2021: https://dictionary.cambridge.org/dictionary/english/conciseness

How to cite this article: Dori D. Model-based standards authoring: ISO 15288 as a case in point. *Systems Engineering*. 2024;27:302–314. https://doi.org/10.1002/sys.21721

AUTHOR BIOGRAPHY

Dov Dori is Life Fellow of IEEE and Fellow of INCOSE, IAPR, and AAIA. He is Professor Emeritus of Systems Engineering and Head of the Enterprise Systems Modeling Laboratory at the Faculty of Data and Decision Sciences, Technion, Israel Institute of Technology. He has been Visiting Professor at MIT intermittently between 1999 and 2020. In 1993, Dr. Dori invented Object-Process Methodology (OPM ISO

19450:2015). He wrote two books on OPM and has been central to the field of model-based systems engineering (MBSE). Prof. Dori has authored over 400 publications and supervised 60 graduate students. He chaired nine international conferences, was Associate Editor of IEEE T-PAMI and Systems Engineering, and Founding Co-Chair of the IEEE Society of Systems, Men, and Cybernetics Technical Committee on MBSE. His 2002 and 2016 books are the basis for the MBSE edX certificate program and MOOC series. He has received various research, innovation, and teaching awards, including INCOSE Pioneer Award, INCOSE Propeller Hat Award, and he is a member of Omega Alpha Association—International Honor Society for Systems Engineering.